Since the start of the term, both of us have started to truly emphasize the importance of looking at the big picture prior to thinking of ways to implement our ideas. By gaining a grasp on exactly what the specification is requiring of us and looking for an end to end solution, we can have more modular code and can avoid facing concerns that we would not have otherwise considered. Instead of simply jumping into implementing our first solution that may work, taking the time to produce a more elegant, thought-out solution pays off in debugging time later. It's also usually easier for other people to understand your logic. This approach is applicable for something as small as writing a helper function. Eventually, however, one needs to start testing out ideas and discovering how things works for dwelling on conceptualization without actually implementing your ideas can often lead to a futile quest for perfection.

Even after you have started implementing, taking a step back and considering whether refactoring your design that you have conceptualized is also important. Sometimes after spending so much time designing, you are reluctant to change aspects of your code. However, no matter how well you think you see the big picture at the beginning, things will change/new problems will arise, and you need to be flexible! This is an approach that we have found useful and hope to use more in the future when we are stuck.